The Future of the Open Trace Format (OTF) and Open Event Trace Recording

SC 2010 BOF Session
New Orleans, Nov 16th 2010

Andreas Knüpfer, Felix Wolf, Michael Gerndt, Sameer Shende

Agenda

- Introduction
- The OTF2 event trace data format
- The Score-P measurement system for parallel performance analysis
- The Score-P Online Analysis Interface
- Connecting the TAU Performance Tools
- Conclusions and Discussion

Introduction

Andreas Knüpfer, ZIH, TU Dresden
andreas.knuepfer@tu-dresden.de

What is OTF?

- Open Trace Format
- Event Tracing Data Format
- Parallel Performance Analysis

- Open Source Trace Format Library and Support Tools
- under BSD Open Source License
- originally developed by ZIH/TU Dresden (Vampir tool group) together with University of Oregon (TAU tool group) with funding by LLNL
- announced at SC'2006 BOF

OTF Features

- store events of interest with specific properties
- Typical events: enter/leave, send/receive, collective communication, hardware performance counters, and more
- Fast and platform independent,
- efficient ASCII encoding, Zlib compression
- fast selective access (for processes and time intervals)
- flexible container for n process using m streams (files)
- read/write APIs for C/C++/Python, support tools
- forward and backward compatible

OTF Adoption

Well received in HPC tools community, e.g. part of:

- VampirTrace, also integrated in Open MPI
- TAU performance tools
- Open|Speedshop
- Sun Performance Analyzer
- Microsoft HPC Pack 2008 SDK

- many more internal/experimental tools/workflows
- from academia and industry

The Open Trace Format Version 2 (OTF2)

Andreas Knüpfer, ZIH, TU Dresden
andreas.knuepfer@tu-dresden.de

Why a new format?

- OTF2 is successor format to OTF and Epilog
- Major re-design and new implementation

- Scalability limits
- Interoperability limits

Scalability Limits

- recording and interactive visualization
  - 10 000 processes/threads and
  - 100 GB to 1 TB of event trace data

- one file per process/thread created during data collection
- reduce (merge) afterwards
- over-stresses all parallel file systems

- exploding representational of process groups (MPI comm.)

- unification of identifiers/definitions requires complete copy

Interoperability Limits

Other event trace data formats very similar

- Epilog (Kojak/Scalasca)
- Paje format (Paje)
- STF (Intel Trace Analyzer)
- Tau trace format (Tau)
- Slog2 (Jumpshot)
- Paraver format (Paraver)

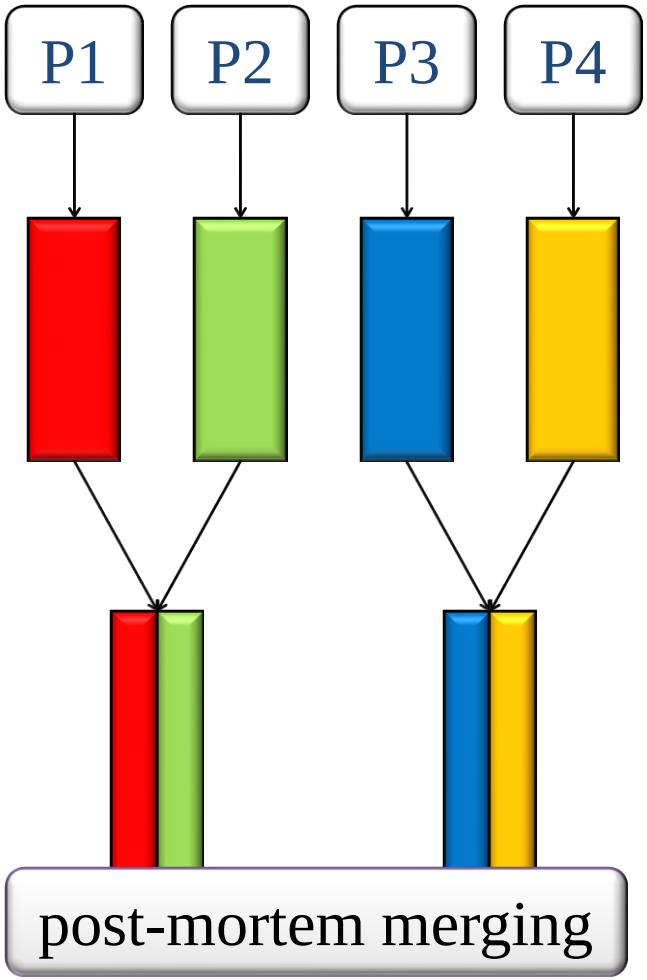
- Inconvenient when using multiple performance tools
- convert formats or re-run measurements

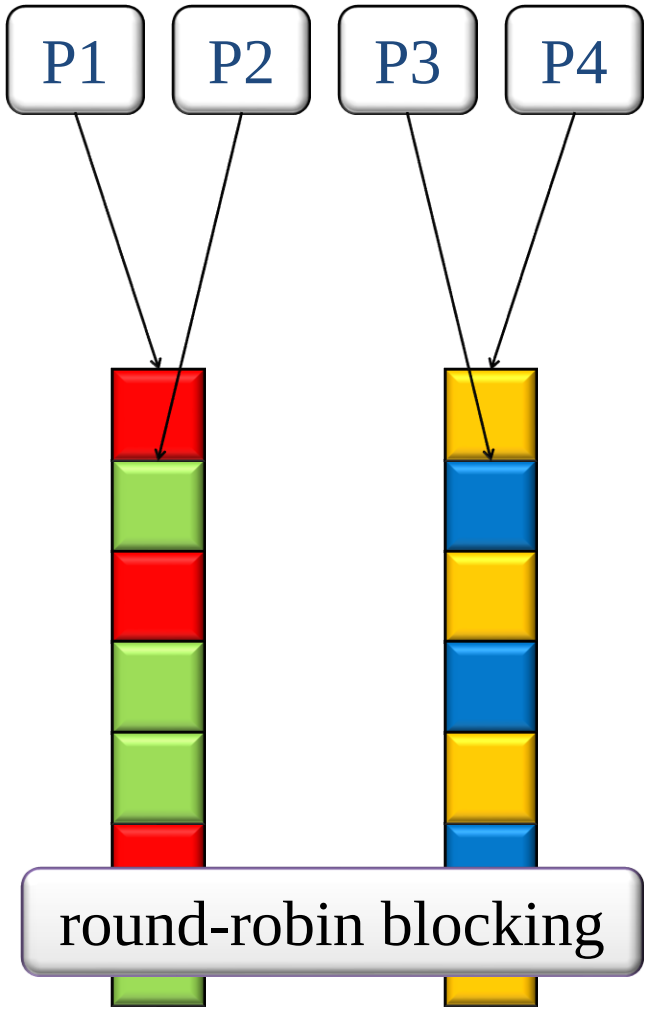
OTF2 Design Goals

- Same basic structure as OTF/Epilog/other formats
- Good read/write performance
- High scalability
- Low overhead (storage space and processing time)
- Integrate OTF2 memory buffer as data recording buffer
- Reduce number of files during initial writing
  via SION Library [1] http://www.fz-juelich.de/jsc/sionlib/
- Compatibility reader for OTF and Epilog formats

[1] Wolfgang Frings, Felix Wolf, Ventsislav Petkov: “Scalable Massively Parallel I/O to Task-Local Files”, in Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, Portland, Oregon, November 14 - 20, 2009, SC'09 Technical paper, Article No. 17, New York, ACM, 2009.ISBN 978-1-60558-744-8. pp. 1 - 11

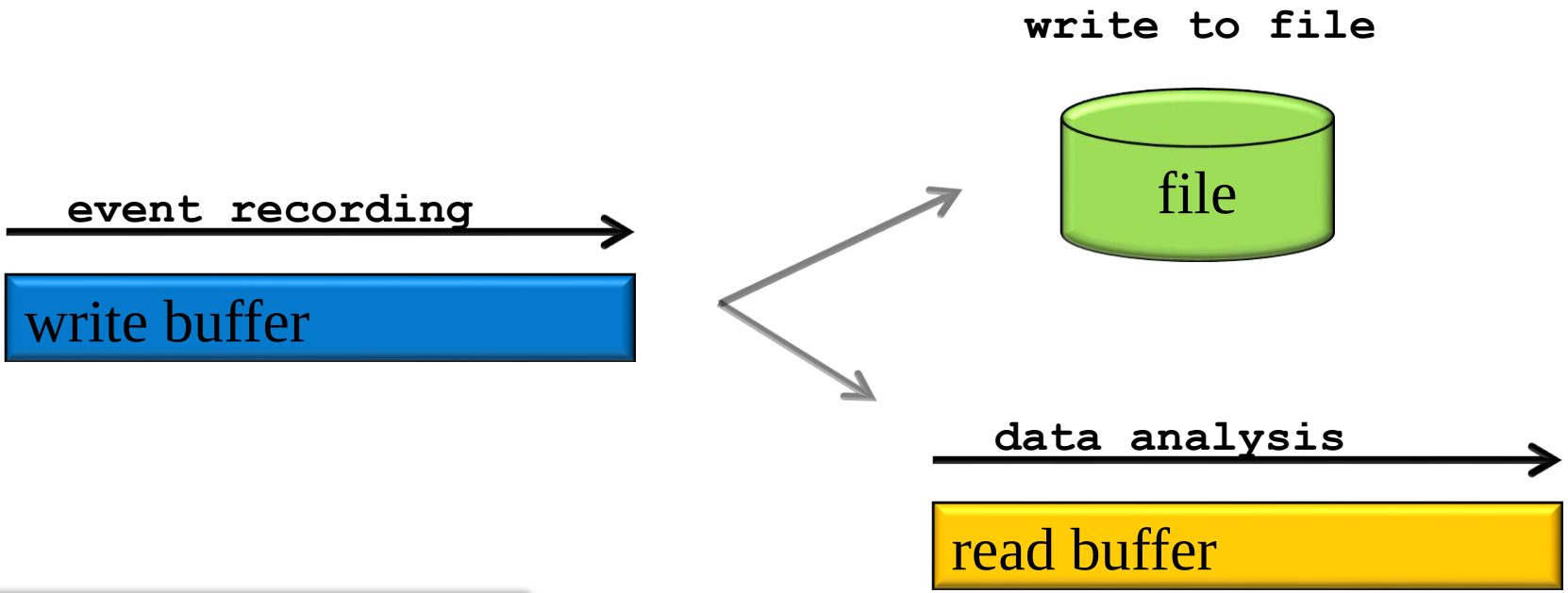
OTF vs. OTF2 File Layout





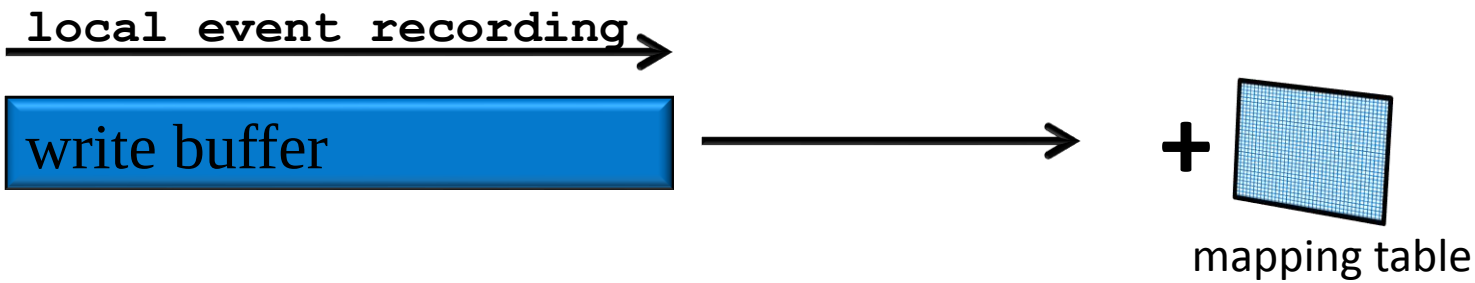
Integration in Measurement System

- OTF2 buffer directly integrated as measurement buffer
- Eventually store to file
- Or re-use as read buffer, avoid I/O altogether

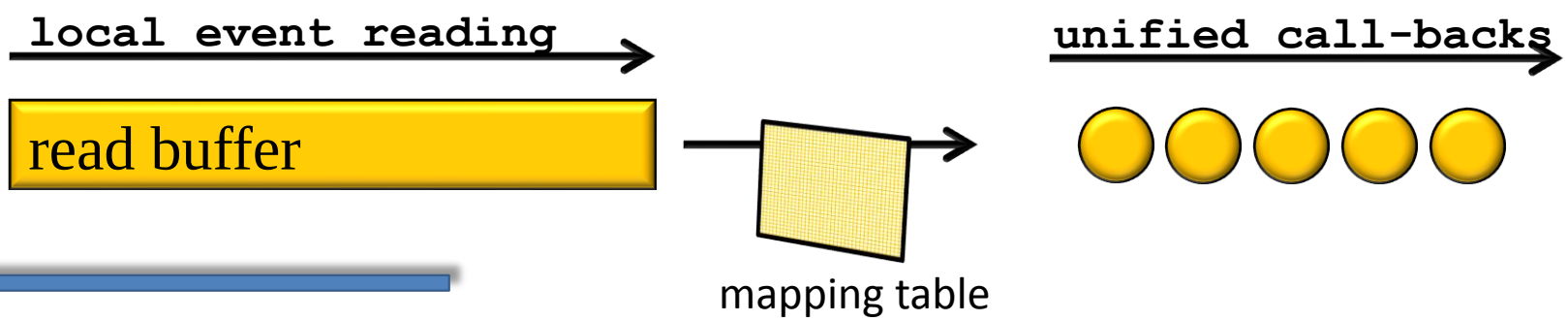


Unification without Copying

- Unification of local IDs and timestamps
- Local-to-global mapping tables, no file copies



- Read library applies mappings on-the-fly
- Read local IDs, translate, deliver global IDs



OTF2 Development

- OTF2 developed as part of the SILC project
- Funded by BMBF/Germany and DOE/USA

- collaboration between:
  - ZIH/TU Dresden (Vampir tool group),
  - JSC/Forschungszentrum Jülich and German Research School for Simulation Sciences (Scalasca tool group)
  - University of Oregon (Tau tool group)
- Open for other tools/groups

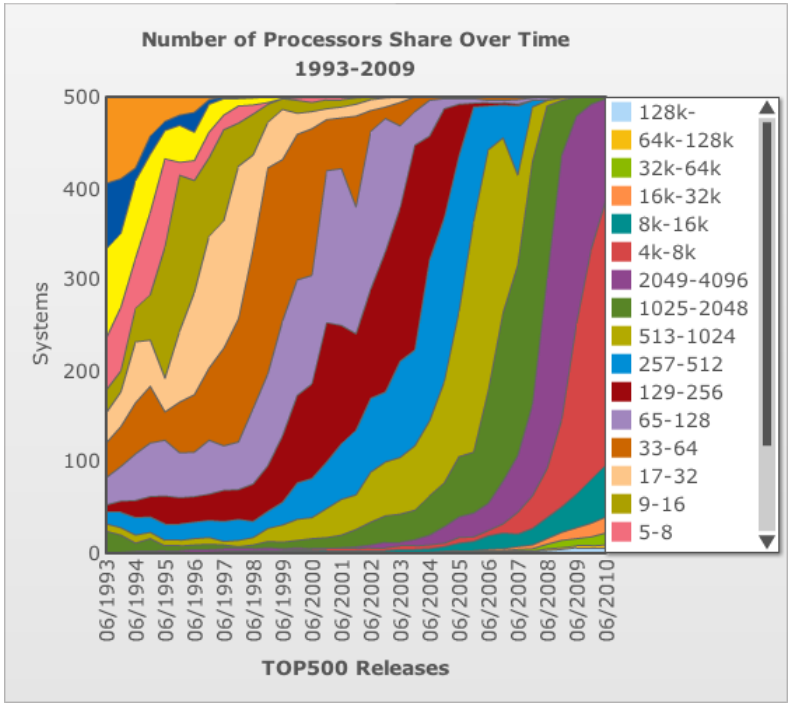
- maintainers: Dominic Eschweiler and Michael Wagner

The Score-P measurement system for parallel performance

Felix Wolf
German Research School for Simulation Sciences

Higher degrees of parallelism

- Optimization of applications more difficult
  - Increasing machine & application complexity
- High demands on scalability of performance tools



Fragmentation of tools landscape

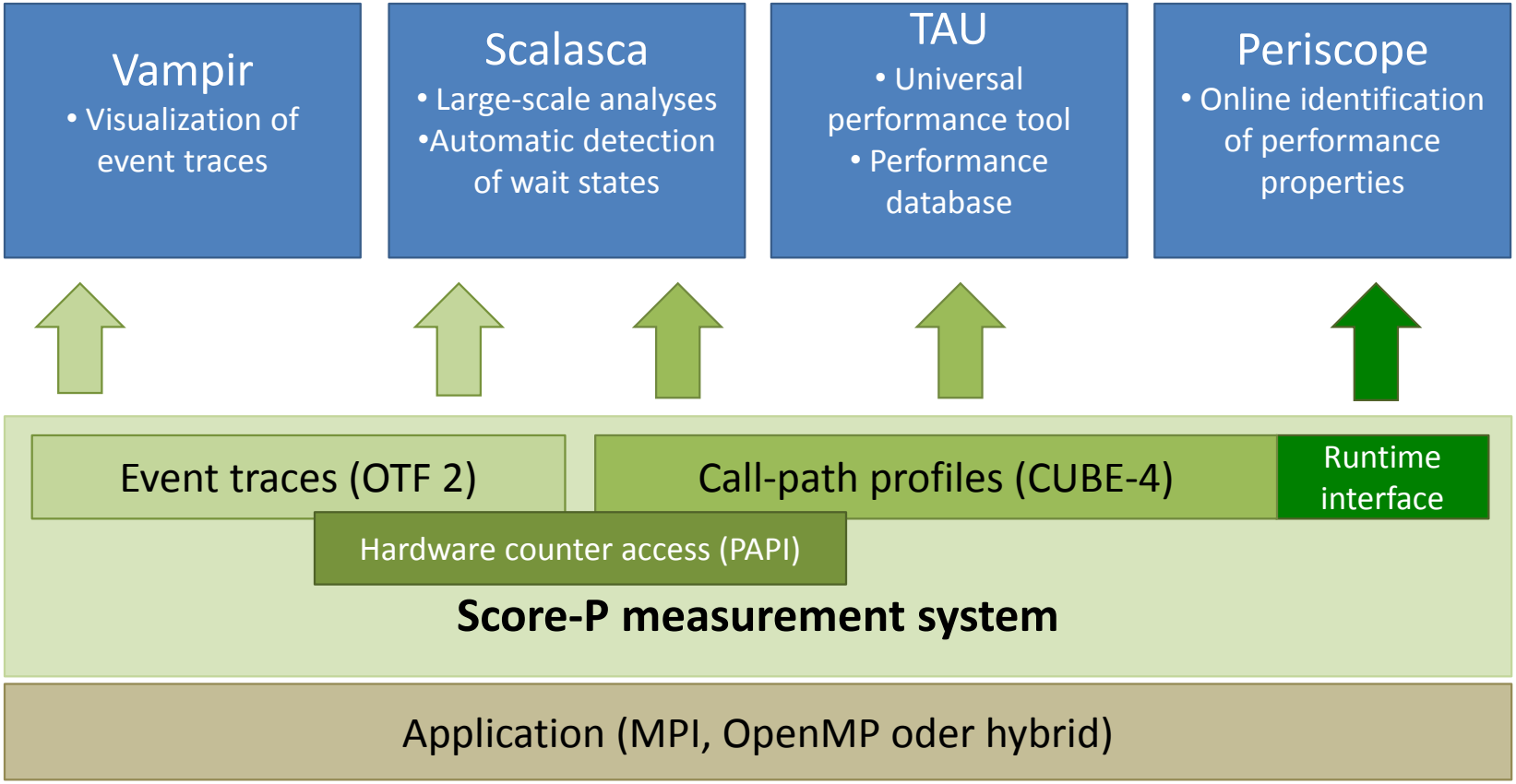
- Several performance tools co-exist
  - With different features
  - But also with overlapping functionality
  - Each comes with its own proprietary measurement system and output formats
    - Limited or expensive interoperability
    - Major effort is spent on their evolution and maintenance
    - Simultaneous installation cumbersome

Vampir	Scalasca	TAU	Periscope
VampirTrace OTF	EPILOG / CUBE	TAU native formats	Online measurement

Objectives

- Make common part a community effort
- Improve interoperability
- Save manpower by sharing resources
- Invest this manpower in analysis functionality
- Simplify joint installation of tools
- Avoid the pitfalls of earlier community efforts
  - Start with small group of partners
  - Build on extensive history of collaboration

A unified performance measurement infrastructure



Score-P partners

- Technische Universität Dresden, Germany
- RWTH Aachen, Germany
- Gesellschaft für numerische Simulation mbH Braunschweig, Germany
- University of Oregon, Eugene, USA
- Forschungszentrum Jülich, Germany
- German Research School for Simulation Sciences, Jülich/Aachen, Germany
- Technische Universität München, Germany

Design goals

- Functional requirements
  - Performance data types: profiles, traces
  - Initially direct instrumentation, later also sampling
  - Offline and online access
  - Metrics: time, communication metrics and hardware counters
  - Initially MPI 2 and OpenMP 3, later also accelerators (CUDA, OpenCL)
- Non-functional requirements
  - Portability: all major HPC platforms
  - Scalability: petascale
  - Low measurement overhead
  - Easy installation through UNITE framework
  - Robustness
  - Open source: New BSD license

Instrumentation

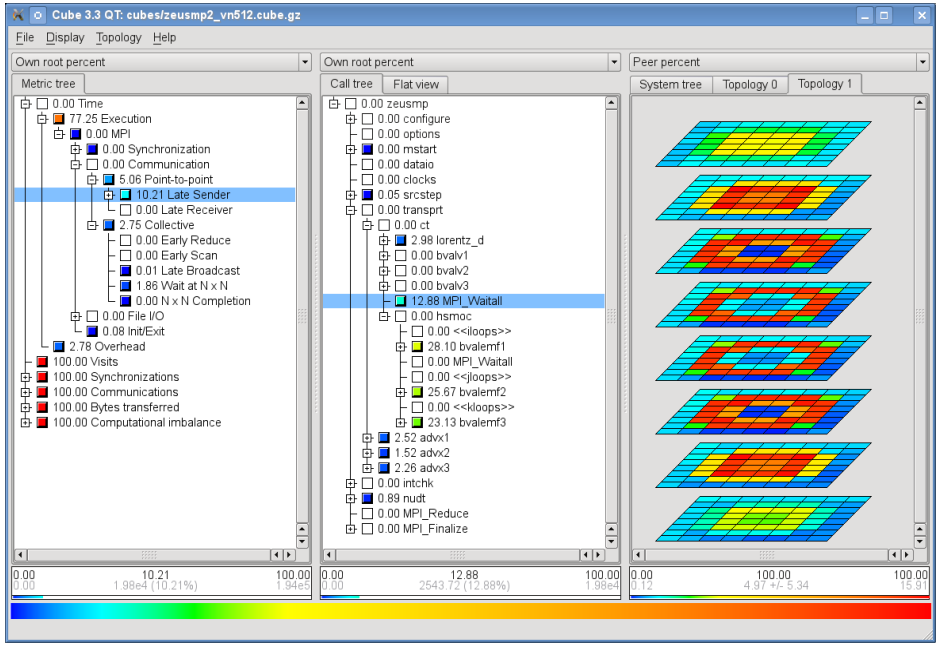
- Compiler instrumentation
- MPI library interposition
- Automatic source-code instrumentation of OpenMP constructs via Opari2
- Manual source code instrumentation
- Automatic source-code instrumentation via TAU instrumentor

CUBE-4

- Latest version of a family of profiling formats
  - Still under development
- Representation of three-dimensional performance space
  - Metric, call path, process or thread
- File organization
  - Metadata stored as XML file
  - Metric values stored in binary format
    - Two files per metric: data + index for storage-efficient sparse representation
- Optimized for
  - High write bandwidth
  - Fast interactive analysis through incremental loading

CUBE-4 cont.

- Will comes with read and write API plus GUI for interactive exploration
- Version 3 available at www.scalasca.org/cube
- CUBE-3 used by
  - Scalasca
  - Marmot
  - ompP
  - TAU



Projects & sponsors

- SILC (2009 – 2011)
  - Initial version
- PRIMA (2009 – 2012)
  - Integration with TAU
- H4H (2010 – 2013)
  - Heterogeneous architectures

Bundesministerium
für Bildung
und Forschung

Status and schedule

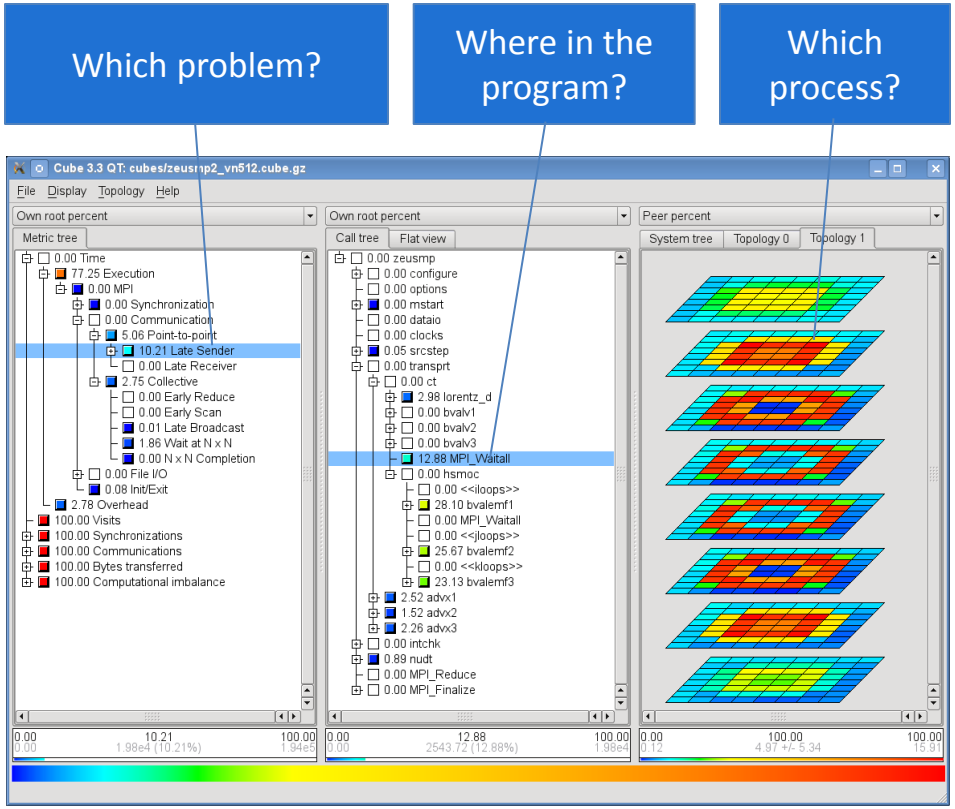
- Initial prototype exists with support for the full tool chain
  - Technology preview available
- First release planned for end of 2011
- Further information
  - www.score-p.org
  - Flyer available in this room

Scalasca

Felix Wolf
German Research School for Simulation Sciences

scalasca

- Performance-analysis toolset for large-scale parallel codes
  - MPI and OpenMP
- Performance overview via call-path profiling
- Identification of wait states via event tracing
- Open source: New BSD
- www.scalasca.org/

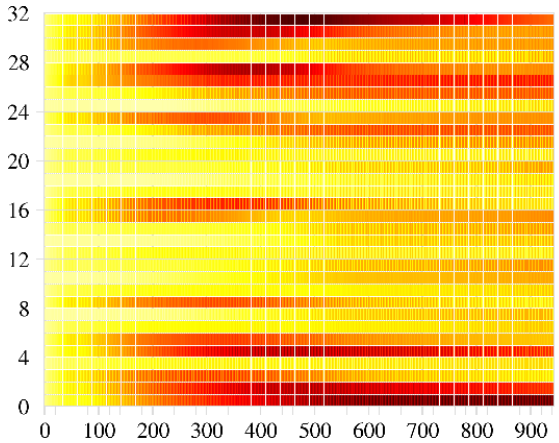


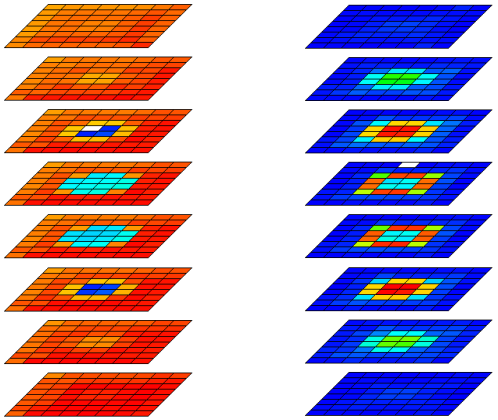
JÜLICH FORSCHUNGSZENTRUM

German Research School for Simulation Sciences

Scalasca directions

- Profiling
  - Analysis of time-dependent behavior
- Tracing
  - Identification of wait states and their root causes
- Higher degrees of scalability
- More info
  - www.scalasca.org
  - JSC booth #1451





SC10 New Orleans, LA

JÜLICH FORSCHUNGSZENTRUM

German Research School for Simulation Sciences

Score-P Online Analysis Interface

Michael Gerndt, TU München

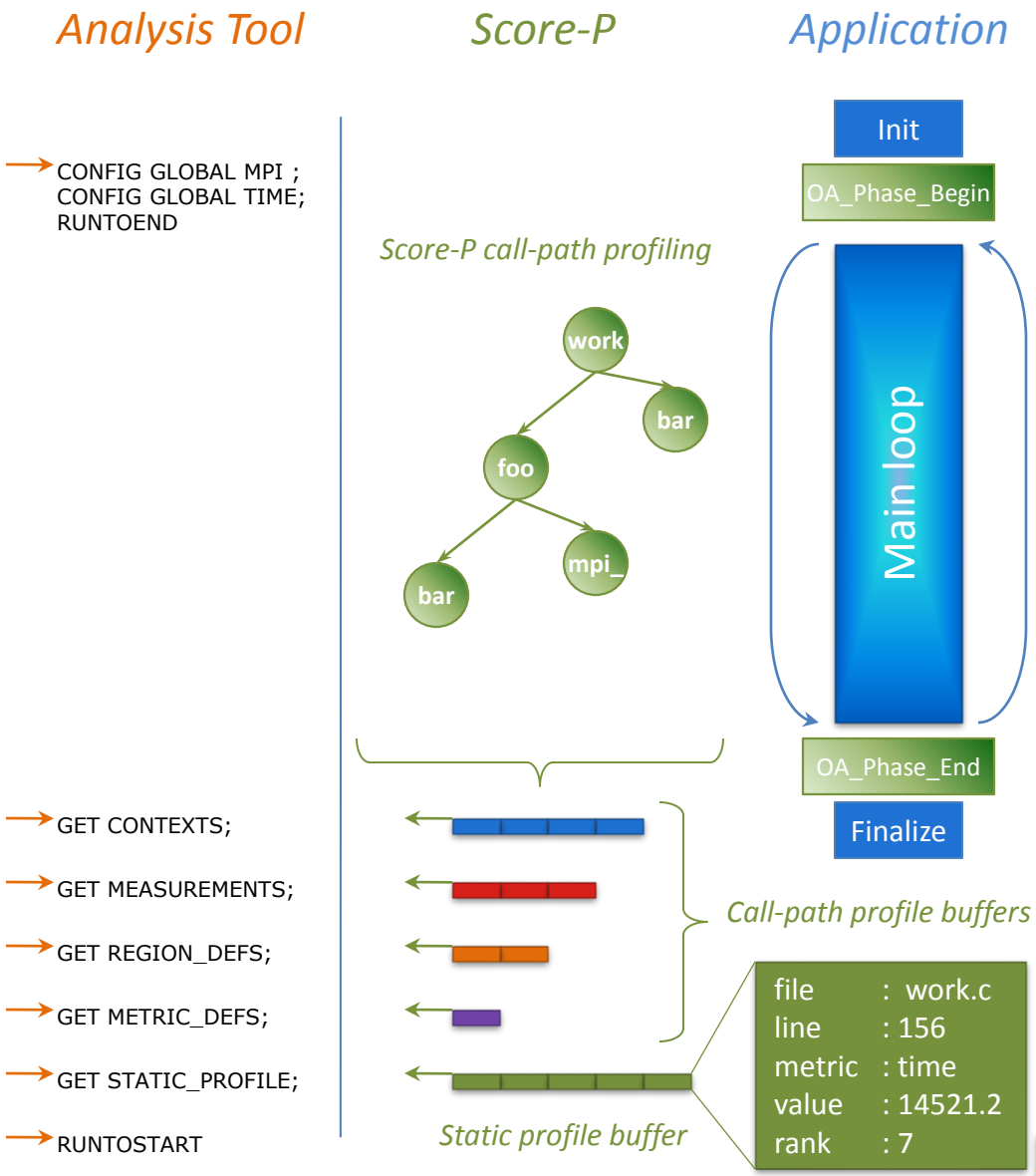
Score-P Online Access Interface

- **Online Access (OA) Interface** enables performance analysis tools to employ the Score-P monitoring infrastructure at runtime remotely over TCP/IP.
- Highlights of **online performance analysis**:
  - Finer grained measurements configuration
  - Multiple performance experiments within one run
  - Remote analysis with measurements acquisition over sockets
  - Faster analysis: one iteration of the time loop could be enough
  - No need to dump measurements into file
- **Features** of the Score-P OA
  - Configure Scope-P
  - Retrieve measured performance data while application is still running
  - Control application execution, allowing multiple measurement iterations
- Score-P OA enables online tools such as **Periscope**

Online Analysis with Score-P

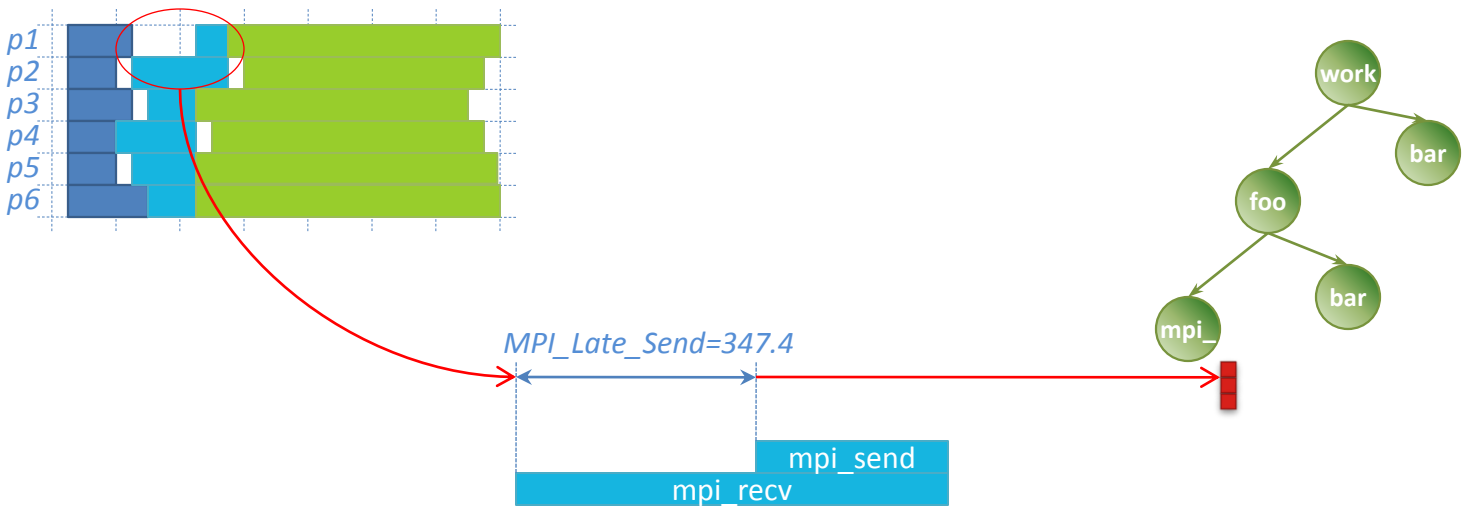
Monitoring Request Interface (MRI)

```
<request_list>      =<request>';'<request>

<request>           =<configuration_request>
                    |<data_request>
                  |<exec_request>

<config_request>  =CONFIG GLOBAL <request>

<request>                    =MPI
                    |<counter_name>
                    |TIME
                    |ALL

<data_request>    =GET <data_type>

<data_type>         =CONTEXTS
                    |MEASUREMENTS
                    |REGION_DEFS
                    |METRIC_DEFS
                    |STATIC_PROFILE

<exec_request>=  TERMINATE
                    |RUNTOSTART
                    |RUNTOEND
```

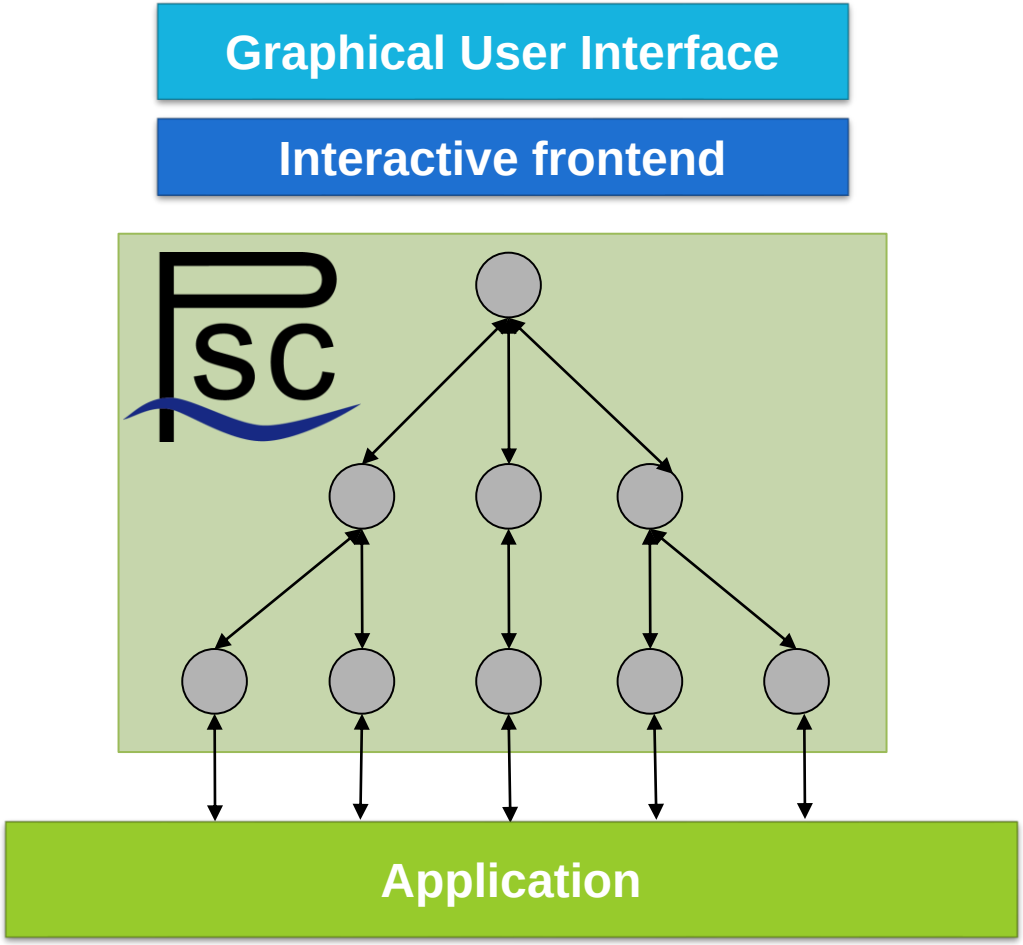


MPI Wait Time profiling

- Automatic detection of MPI wait time patterns
- On-the-fly measurement
- No tracing required
- Nx1, 1xN, NxN, blocking, non-blocking



Periscope on top of Score-P



Eclipse-based GUI

Analysis control

Agent network

Score-P Online Access Interface

TAU Performance System®

Sameer Shende, University of Oregon

TAU Performance System

- Comprehensive Profiling and Tracing
  - Widely ported to all HPC systems
  - CUDA, OpenCL
  - C, C++, Fortran, Chapel, UPC, Python, Java (JVMTI)
  - MPI, OpenMP, Pthread, SHMEM, combination…
- Instrumentation, Measurement, and Analysis
  - Interfaces with Score-P, Scalasca, VampirTrace, PAPI
  - Supports binary rewriting (tau_run), automatic source instrumentation (tau_f90.sh), runtime preloading (tau_exec)

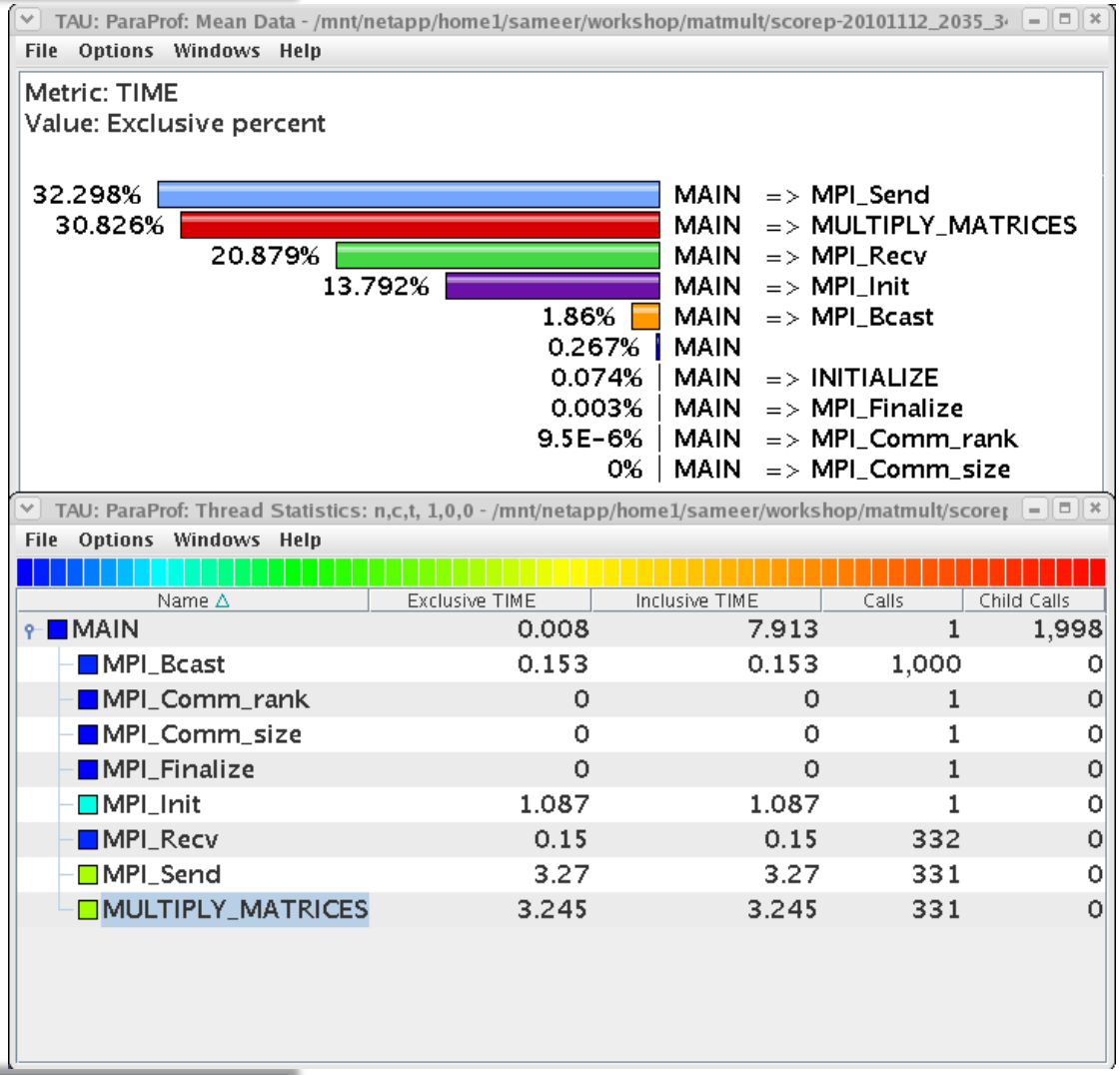
Motivation

- Quest for improved runtime efficiency
  - TAU uses tables based lookup of callpath entities
  - Score-P provides a more efficient tree based data structure
  - Reduces runtime dilation
  - Allows TAU to generate traces that do not require merging or conversion
  - Scalable tracing solutions with Vampir

Usage

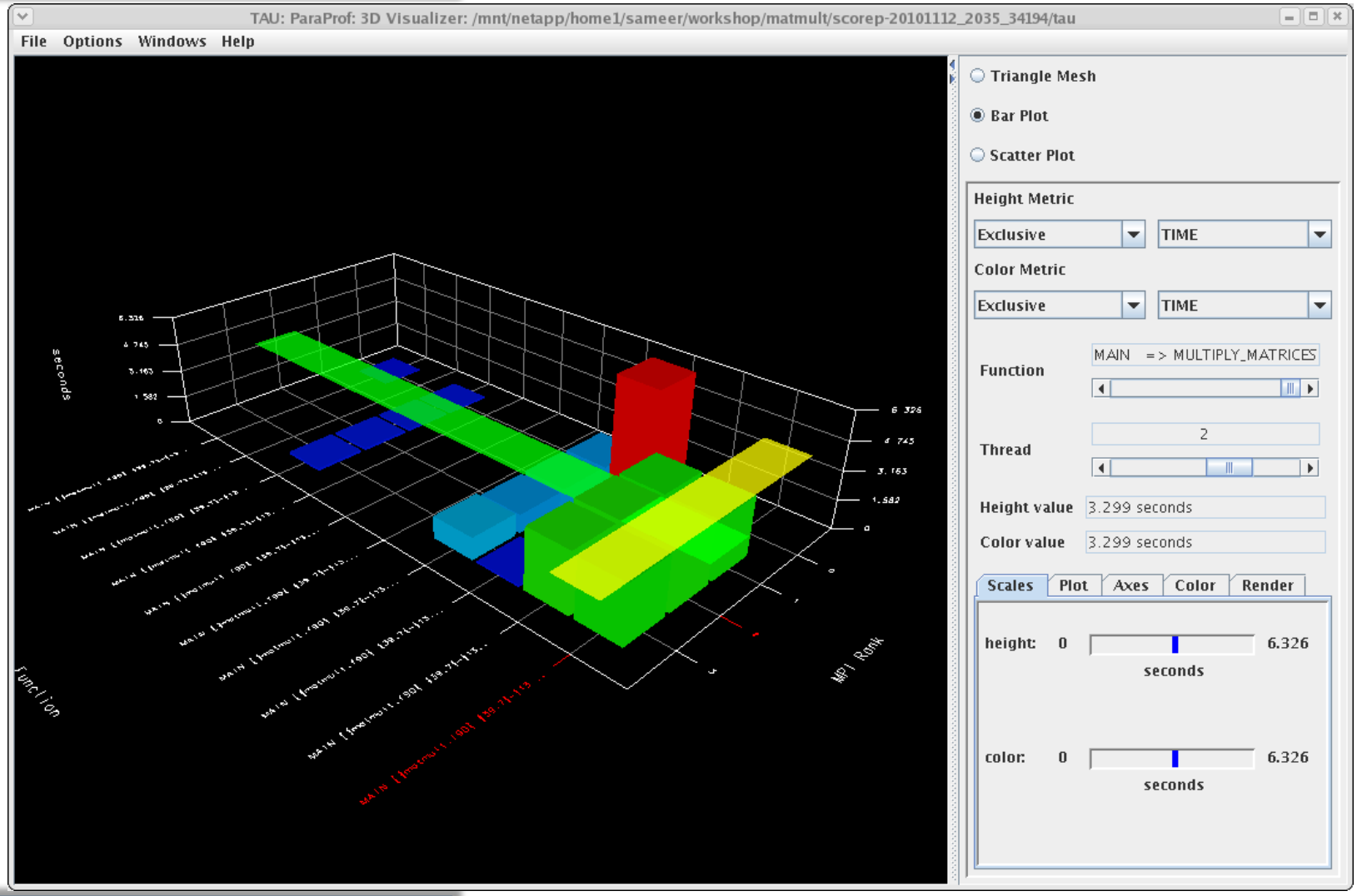
```
> setenv TAU_MAKEFILE $TAU/Makefile.tau-mpi-pdt-scorep
> make F90=tau_f90.sh CC=tau_cc.sh CXX=tau_cxx.sh
> setenv SCOREP_ENABLE_PROFILING 1
> setenv SCOREP_ProfileFormat TauSnapshot
> mpirun -np 4 ./matmult
 Process            2  of            4  is active
 Process            1  of            4  is active Process            3  of
      4  is active

 Process            0  of            4  is active
 c(        1000 ,        1000 ) =    500500000.00000000
> ls -l scorep-20101112_2035_34194/
total 20
drwxr-xr-x 2 sameer paraducks 4096 Nov 12 20:35 tau
drwxr-xr-x 2 sameer paraducks 4096 Nov 12 20:35 traces
-rw-r--r-- 1 sameer paraducks 7225 Nov 12 20:35 traces.def
-rw-r--r-- 1 sameer paraducks  182 Nov 12 20:35 traces.otf2
> cd scorep-20101112_2035_34194/tau/
> paraprof -f snapshot &
```

Generating TAU data with Score-P



Name	Exclusive TIME	Inclusive TIME	Calls	Child Calls
MAIN	0.008	7.913	1	1,998
MPI_Bcast	0.153	0.153	1,000	0
MPI_Comm_rank	0	0	1	0
MPI_Comm_size	0	0	1	0
MPI_Finalize	0	0	1	0
MPI_Init	1.087	1.087	1	0
MPI_Recv	0.15	0.15	332	0
MPI_Send	3.27	3.27	331	0
MULTIPLY_MATRICES	3.245	3.245	331	0

TAU's Paraprof 3D Profile Browser



Status and release schedule

- TAU v2.20 released at SC’10 has full support for Score-P measurement
- ./configure –scorep=<dir> ; make install

And use $TAU/Makefile.tau-*scorep*

TAU Acknowledgements

- Department of Energy (DOE)
  - Office of Science
    - MICS, Argonne National Lab
  - ASC/NNSA
    - University of Utah ASC/NNSA Level 1
    - ASC/NNSA, Lawrence Livermore National Lab
- Department of Defense (DoD)
  - HPC Modernization Office (HPCMO)
- NSF SDCI and SI2 programs
- Research Centre Juelich
- ANL, ORNL, LLNL, LANL, SNL, PNNL
- TU Dresden
- ParaTools, Inc.

Conclusions and Discussion

Andreas Knüpfer, ZIH, TU Dresden
andreas.knuepfer@tu-dresden.de

Score-P Technology Demonstration

- First announcement of next generation tools infrastructure
- More information and future updates at

http://www.score-p.org

- Technology preview tarball available
  - Early functional version, try with SMG2K benchmark
  - For interested HPC tool developers
  - Not yet recommended for users or productive use

Discussion

Thank you for your attention!

Questions?

Comments?